哈尔滨工业大学计算机科学与技术学院

实验报告

课程名称：机器学习

课程类型：必修

实验题目：实现k-means聚类和混合高斯模型

学号：1171000820

姓名：陈嵩

一、实验目的

实现一个k-Means算法和混合高斯模型，并用EM算法估计模型中的参数。

二、实验要求及实验环境

实验要求

测试

用高斯分布产生k个高斯分布的数据(不同均值和方差)(其中参数自己设定)

1. 用k-Means聚类测试效果，
2. 用混合高斯模型和你实现的EM算法估计参数，看看每次迭代后似然值变化情况，考察EM算法是否可以获得正确结果(与你的设定结果比较)。

应用

可以在UCI上找一个简单问题数据，用你实现的GMM进行聚类。

实验环境

- python 3.7.1

三、设计思想(本程序中用到的主要算法及数据结构)

1.算法原理

1.1 K-Means算法原理

给定训练样本$X = x_1, x_2, \ldots, x_m$和划分聚类的数量$k$，给出一个簇划分$C = C_1, C_2, \ldots, C_k$，使得该划分的**平方误差**$E$最小化，有式$(1)$

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||_2^2 \tag{1}$$

其中，$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$,$\mu_i$是簇$C_i$的均值向量。$E$刻画了簇内样本围绕簇的均值向量的紧密程度，值越小，则簇内样本的相似度越高。

具体迭代过程如下：

1. 根据输入的超参数K首先初始化一些向量（可以从现有的向量中挑选），作为各簇的均值向量。
2. 根据初始化的均值向量给出训练样本的一个划分，计算各个训练样本到各个均指向量的距离，找出距离最近的均值向量，并将该样本分至该均值向量所代表的簇。
3. 根据新的簇划分，重新计算每个簇的均值向量，如果新的均值向量与旧的均值向量相同，认为算法收敛；否则，更新均值向量，回到第2步.重新迭代求解。

1.2 GMM模型

多元高斯分布生成的n维随机变量x的密度函数为：

$$p(x|\mu,\Sigma)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{2}$$

其中μ为n维的均值向量，Σ为$n\times n$的协方差矩阵。

给定训练样本集$X=\{x_1,x_2,\ldots,x_m\}$，我们认为它是一个$N*d$维的矩阵，N为样本数量，d为单个样本的维度数量。

对于一个样本数据x，我们可以认为它是由多个对应维度的多元高斯分布所生成，所以，我们采用混合高斯模型来表征数据，其定义为：

$$p_{\mathcal{M}}(x)=\sum_{i=1}^{k}\alpha_i p(x|\mu_i,\Sigma_i) \tag{3}$$

我们认为数据由k个高斯分布混合生成，每个高斯分布对应一个混合成分。其中μ_i,Σ_i是第i个高斯分布的均值和协方差矩阵，$\alpha_i>0$为相应的混合系数，满足$\sum_{i=1}^{k}\alpha_i=1$。因此，我们也可以认为该数据的生成相当于从k个高斯分布中挑选出一个所生成，我们设K维01变量z来表征对应的样本数据x所属的类别，即由哪个高斯分布所生成，满足$\sum\limits_{k}z_k=1$,即仅有一维为1，则α_i加权平均概率值可以表征z的分布。

而z_j的后验分布为：

$$\gamma(z_k)=p_{\mathcal{M}}(z_k=1|x_j)=\frac{p(z_k=1)\cdot p_{\mathcal{M}}(x_j|z_k=1)}{p_{\mathcal{M}}(x_j)}=\frac{\alpha_i p(x_j|\mu_i,\Sigma_i)}{\sum\limits_{l=1}^{k}\alpha_l p(x_j|\mu_l,\Sigma_l)} \tag{4}$$

$\gamma(z_k)$代表了样本x_j由第k个高斯混合分布生成的后验概率。

当式(4)已知时，混合高斯模型将训练样本X划分成了K个簇$C=\mathbf{C_1},\mathbf{C_2},\ldots,\mathbf{C_K}$，对于每一个样本$x_i$，其类别为$k$，满足：

$$k=\arg\max_{k}\gamma(z_k) \tag{5}$$

也就是说，选择后验概率最大的类别，划分样本的类别。

因此，当我们观测到样本集X时，我们可以可以采用极大似然估计来求解样本的类别分布：

$$\ln p(X|\alpha,\mu.\Sigma)=\ln\left(\prod_{n=1}^{N}p_{\mathcal{M}}(x_n)\right)=\sum_{n=1}^{N}\ln\left(\sum_{k=1}^{K}\alpha_k p(x_n|\mu_k,\Sigma_k)\right) \tag{6}$$

使式(6)最大化，对μ_k求导令导数为0有：

$$\sum_{n=1}^{N}\frac{\alpha_i p(x_n|\mu_k,\Sigma_k)}{\sum\limits_{k=1}^{K}\alpha_k p(x_n|\mu_k,\Sigma_k)}\Sigma_i^{-1}(x_n-\mu_k)=0 \tag{7}$$

化简有：

$$\gamma(z_{nk}) = \frac{p_{\mathcal{M}}(z_k = 1|x_n)}{\sum_{k=1}^{K} p_{\mathcal{M}}(z_k = 1|x_n)} \tag{8}$$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{N_k}$$

这里我们可以认为，对于第k个分布，它的均值为，按照概率$\gamma(z_{nk})$分类为第k类的样本的加权平均值，N_k即是分为该类的样本的概率总值。

同理式(6)对Σ_k求导令导数为0有：

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{N_k} \tag{9}$$

对于混合系数α_k，还需要满足$\alpha_k \geq 0, \sum_{k=1}^{K} \alpha_k = 1$。因此，我们在式$(6)$的基础上增加拉格朗日项：

$$\ln p(X|\alpha, \mu. \Sigma) + \lambda \left(\sum_{k=1}^{K} \alpha_k - 1 \right) \tag{10}$$

其中λ为拉格朗日乘子，式(10)对α_k求导，令导数为0，有：

$$\sum_{n=1}^{N} \frac{p(x_n|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \alpha_k p(x_n|\mu_k, \Sigma_k)} + \lambda = 0 \tag{11}$$

式(11)两边同乘α_k并将$k \in \{1, 2, \ldots, K\}$代入相加得：

$$\sum_{k=1}^{K} \gamma(z_{nk}) + \lambda \sum_{k=1}^{K} \alpha_i = 0 \tag{12}$$

整理一下，代入$\sum_{i=1}^{k} \alpha_i = 1$:

$$N + \lambda = 0 \tag{13}$$

从而有$\lambda = -N$，代入式(11)有：

$$\alpha_k = \frac{N_k}{N} \tag{14}$$

2.算法的实现

2.1 K-Means算法实现

2.1.1 随机选择样本作为初始均值向量

1. 从训练样本X中随机选择k个样本作为初始的均值向量$\mu_1, \mu_2, \ldots, \mu_k$
2. 重复迭代直到算法收敛：
    1. 初始化各簇C_1，$C_2, \ldots, C_3$，将对应的均值向量加入其中。
    2. 对每一个样本x_i，计算其与每一个均值向量μ_j的距离。
    3. 将样本x_j划分到相应的簇C_k,k满足$\arg\min_k \ \|x_i - \mu_k\|$，即划分至与样本最近的均值向量所属的簇中。
    4. 计算当前状态下每个簇的均值向量$\hat{\mu_i} = \frac{1}{|C_i|} \sum_{x \in C_i} x$

5. 如果对于所有的$i \in 1, 2, \ldots, k$，均有$\hat{\mu_i} = \mu_i$，则终止迭代；
6. 否则更新均值向量，更新$\mu_i = \hat{\mu_i}$，返回第1步重新进行迭代。

2.2 EM算法求解GMM模型

GMM常采用EM算法进行迭代优化求解，其中每次迭代中，先计算当前参数下每个样本由每个高斯分布生成的后验概率，即“E步”，这里的E即为期望的意思，实际上是经过引入$\gamma(z_{nk})$后验概率后，使用最大似然估计所得的函数，将其视为一个期望值，利用Jenson不等式，从而得到似然值有关于$\gamma(z_{nk})$后验概率的一个下界，并得到新的$\gamma(z_{nk})$

而对于该下界，再根据新的$\gamma(z_{nk})$，按照式$(8)(9)(14)$更新参数，所谓“M步”，这里的M为最大化的意思，目的是找到新的参数的值使得$\gamma(z_{nk})$函数最大，进而逼近样本和分类关系的后验概率的下界，更好的估计该模型的隐含参数。

给定训练样本X和高斯混合成分的数目k。

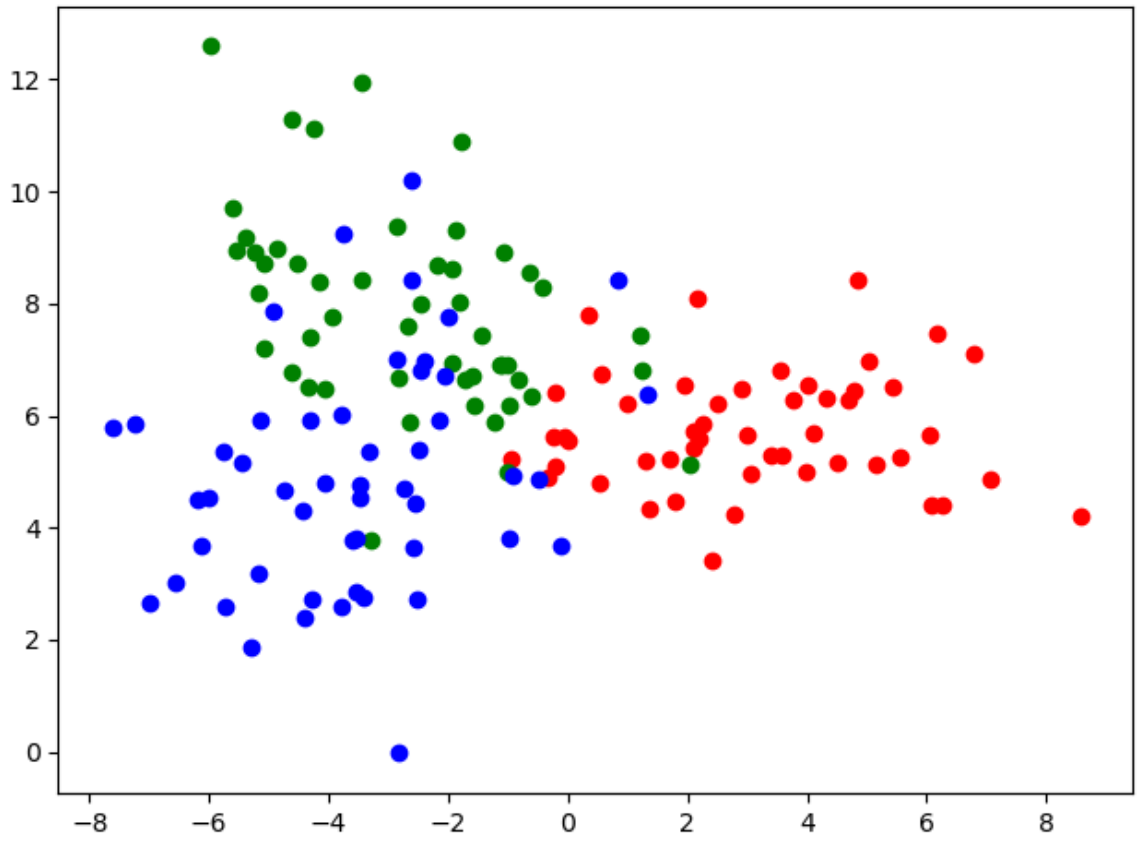
1. 随机初始化参数$\alpha_i, \mu_i, \Sigma_i | i \in 1, 2, \ldots, k$
2. 开始迭代至带到迭代次数或者是参数值不再发生变化：
    1. E步，根据式(4)计算每个样本由各个混合高斯成分生成的后验概率
    2. M步，根据式$(8)(9)(14)$更新参数$\alpha_i, \mu_i, \Sigma_i | i \in 1, 2, \ldots, k$
3. 根据式(5)确定每个样本的类别k，并将其加入相应的簇C_k
4. 输出各个簇内的样本划分。

四、实验结果分析

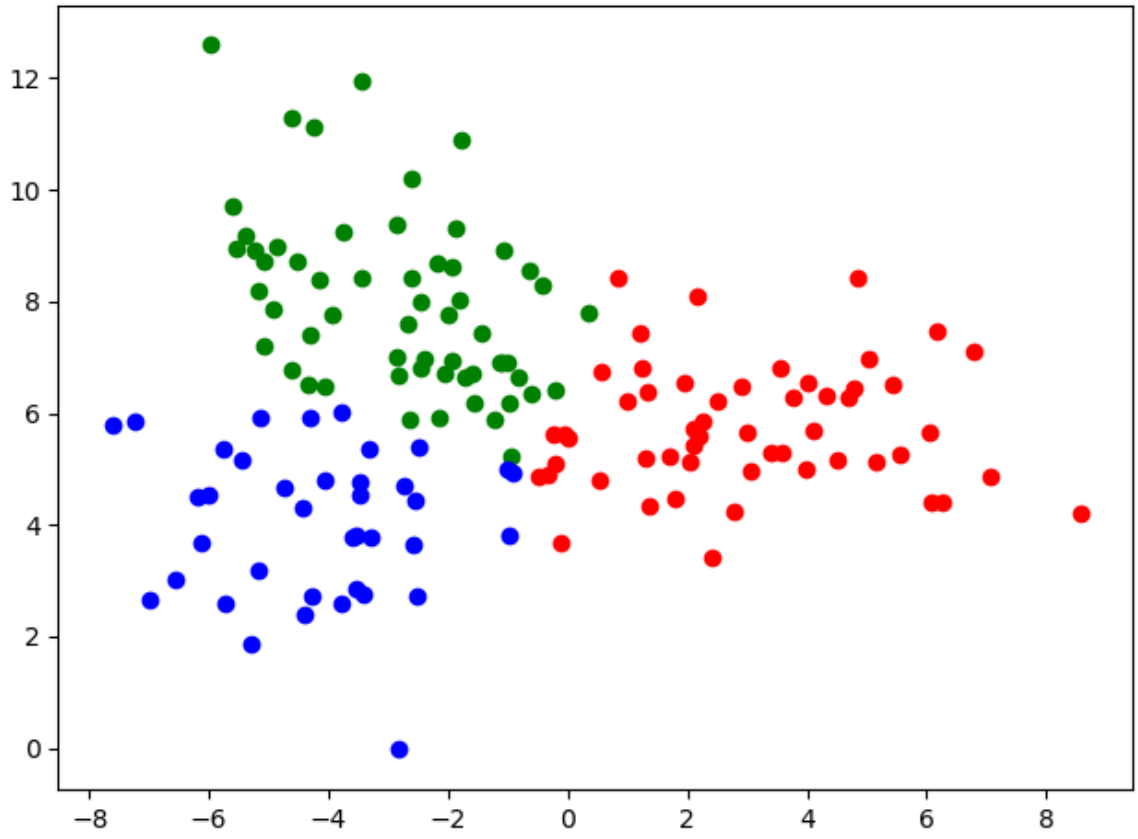
1. 生成数据的测试

生成K=3的2维数据测试K-Means和GMM的效果，各高斯分布的均值和协方差矩阵均不同且随机生成。

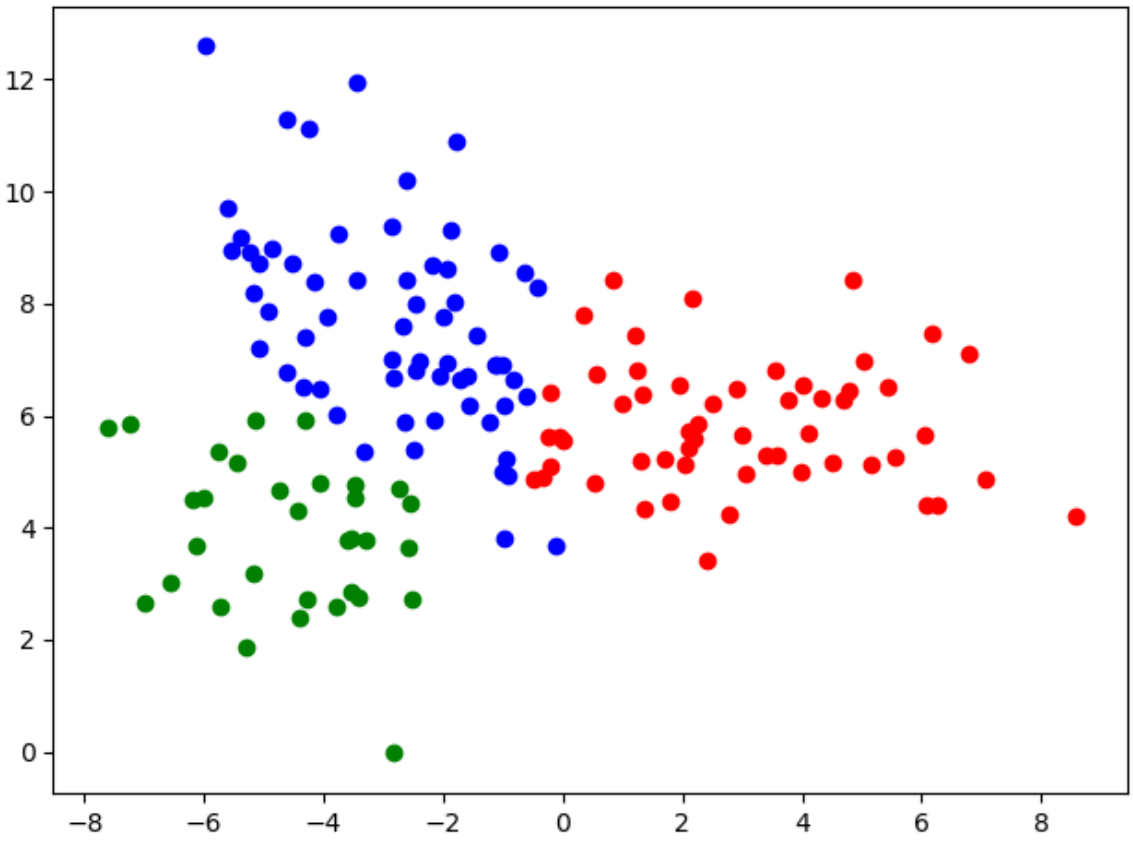
1.1 聚类效果对比



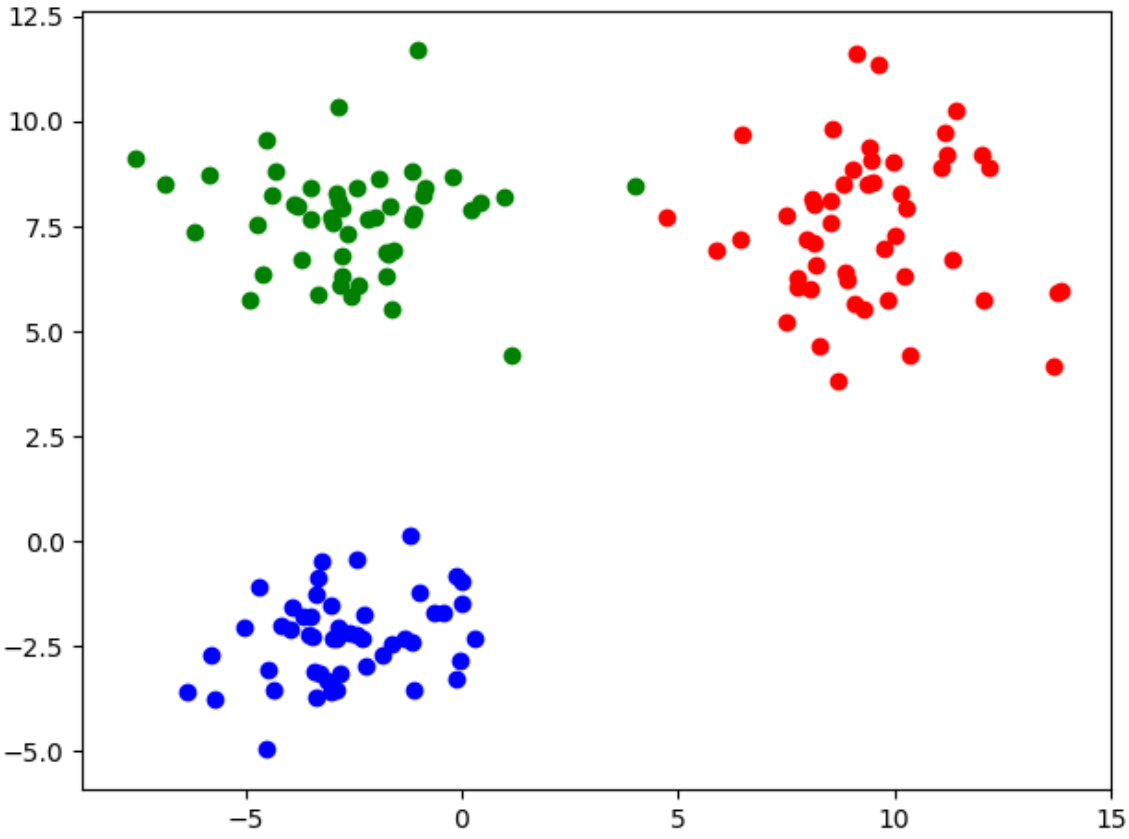
上图为生成的随机数据，数据分布较集中，各个类别有很多交叉。



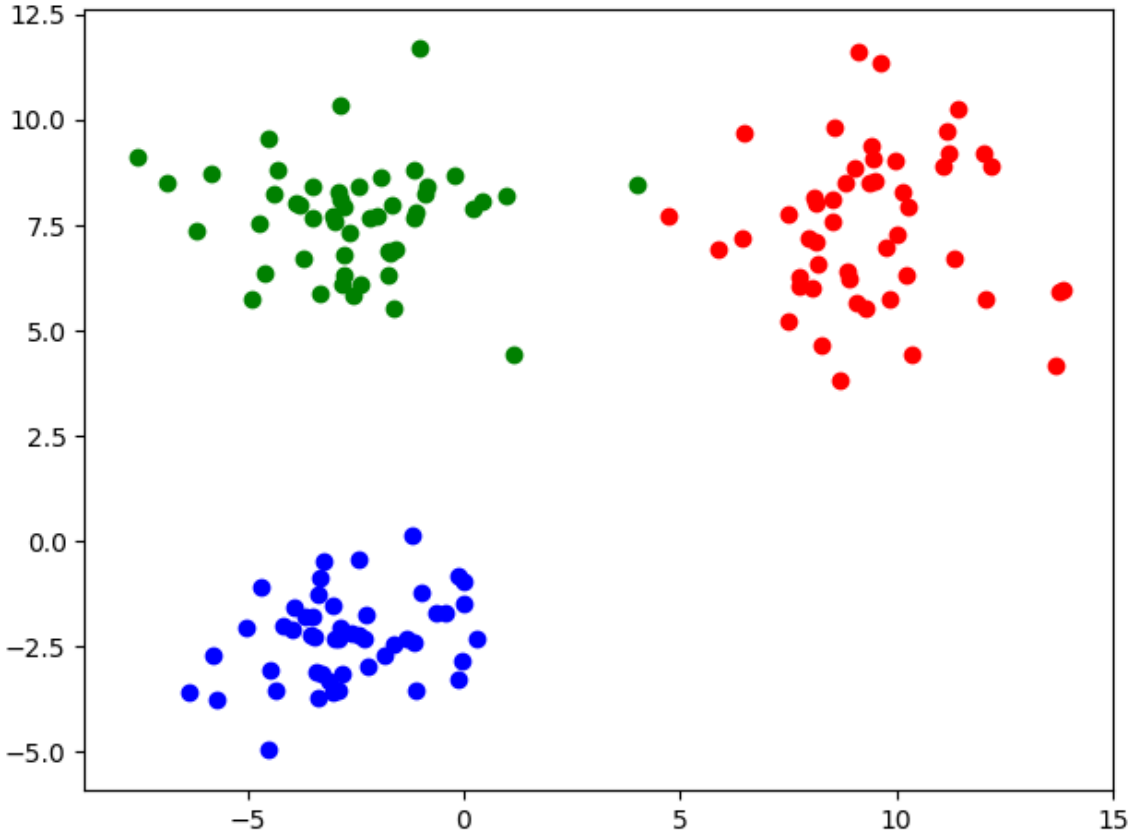
上图为K-Means聚类效果。



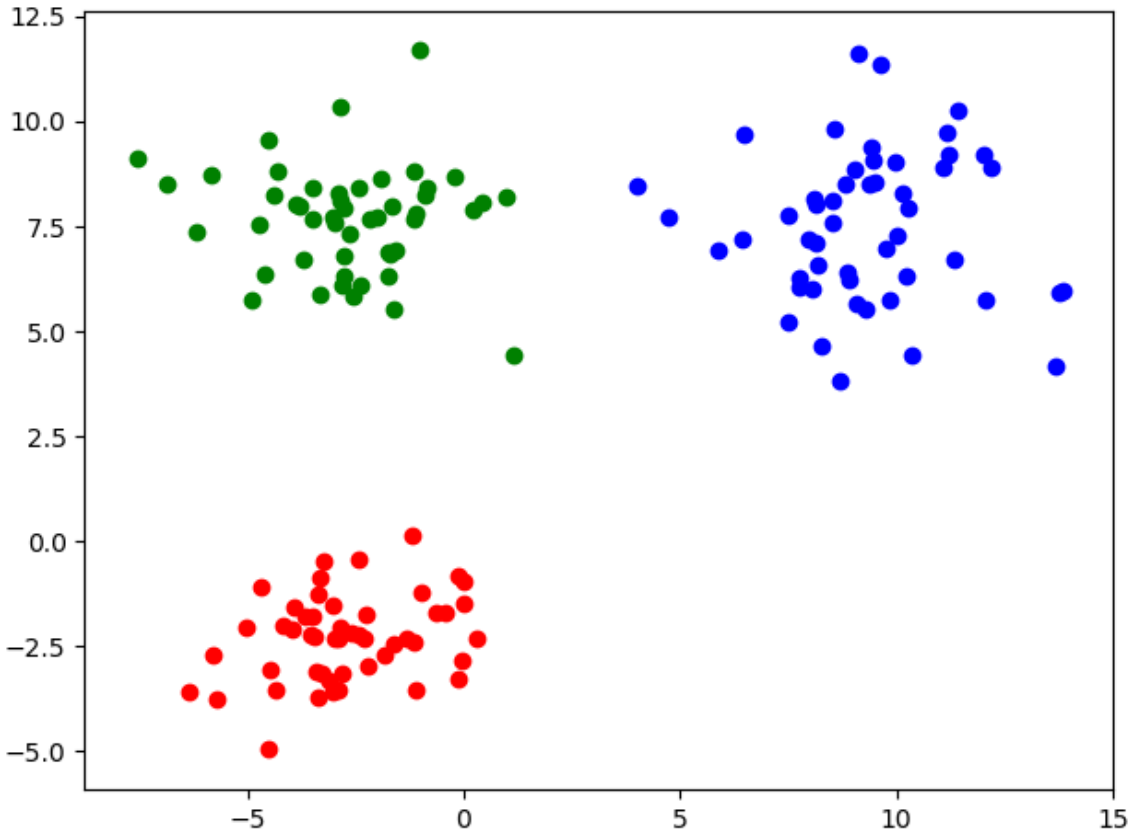
上图为EM聚类效果。



上图为第二次生成的数据



上图K-Means聚类效果



上图为EM聚类效果

可以看到，两种方法在生成数据上的表现类似，都可以实现聚类。对于分布较为分散，没有交叉的数据，聚类效果更好。

另外，对于EM算法，如下图

```
GMM epoch:  30  likelihood:  -745.6330426864656
GMM epoch:  60  likelihood:  -745.6323123960525
GMM epoch:  88  likelihood:  -745.6323123955235
```

似然值不断增大，符合预设。

2. UCI数据测试

使用的UCI的数据是Iris(鸢尾花)数据集，根据其4个属性：

- 花萼长度
- 花萼宽度
- 花瓣长度
- 花瓣宽度

预测鸢尾花属于(Setosa, Versicolour, Virginica)三类中的哪一类别。

多次运行EM算法取最优值有：

```
GMM epoch:  30  likelihood:  -287.1503156220204
GMM epoch:  60  likelihood:  -286.37988498078204
GMM epoch:  90  likelihood:  -282.98115987370386
GMM epoch:  101  likelihood:  -282.981159873542
GMM end
```

似然不断增大。

最好效果为正确分类145个样本，准确率为96.7%

五、结论

- K-Means算法对于数据的分布假设较为简单，在复杂的情况下效果不佳。
- K-Means的聚类效果，依赖于类别均值向量的初始化，容易陷入局部最优解。
- K-Means假设使用的欧式距离来衡量样本与各个簇中心的相似度，距离衡量方式较为简略。
- K-Means需要选择合适的超参K，而实际应用中K的选择需要调整。
- 但是超参数量少，且计算简单，收敛较快，可以多次运行取最优解。
- GMM对于数据的假设较为详细，多个高斯分布对于数据拟合效果较好。
- GMM常使用EM算法迭代优化进行参数估计。
- K-Means算法本质上也是一种特殊的EM算法。
- EM算法的核心在于通过不停的调整下界来逼近似然估计，从而最终确定隐变量的取值。

六、参考文献

- Christopher Bishop. Pattern Recognition and Machine Learning.
- 周志华 著. 机器学习, 北京: 清华大学出版社, 2016.1
- UCI Iris

七、附录:源代码(带注释)

见压缩包代码附件。